



Reactive Collision Avoidance Algorithm

Algorithm is used for safe operation of autonomous, collaborative, vehicle formations.

NASA's Jet Propulsion Laboratory, Pasadena, California

The reactive collision avoidance (RCA) algorithm allows a spacecraft to find a fuel-optimal trajectory for avoiding an arbitrary number of colliding spacecraft in real time while accounting for acceleration limits. In addition to spacecraft, the technology can be used for vehicles that can accelerate in any direction, such as helicopters and submersibles.

In contrast to existing, passive algorithms that simultaneously design trajectories for a cluster of vehicles working to achieve a common goal, RCA is implemented onboard spacecraft only when an imminent collision is detected, and then plans a collision avoidance maneuver for only that host vehicle, thus preventing a collision in an off-nominal situation for which passive algorithms cannot. An example scenario for such a situation might be when a spacecraft in the cluster is approaching another one, but enters safe mode and begins to drift. Functionally, the RCA detects colliding spacecraft, plans an

evasion trajectory by solving the Evasion Trajectory Problem (ETP), and then recovers after the collision is avoided. A direct optimization approach was used to develop the algorithm so it can run in real time.

In this innovation, a parameterized class of avoidance trajectories is specified, and then the optimal trajectory is found by searching over the parameters. The class of trajectories is selected as "bang-off-bang" as motivated by optimal control theory. That is, an avoiding spacecraft first applies full acceleration in a constant direction, then coasts, and finally applies full acceleration to stop.

The parameter optimization problem can be solved offline and stored as a look-up table of values. Using a look-up table allows the algorithm to run in real time. Given a colliding spacecraft, the properties of the collision geometry serve as indices of the look-up table that gives the optimal trajectory. For multiple colliding spacecraft, the set of tra-

jectories that avoid all spacecraft is rapidly searched on-line.

The optimal avoidance trajectory is implemented as a receding-horizon model predictive control law. Therefore, at each time step, the optimal avoidance trajectory is found and the first time step of its acceleration is applied. At the next time step of the control computer, the problem is re-solved and the new first time step is again applied. This continual updating allows the RCA algorithm to adapt to a colliding spacecraft that is making erratic course changes.

This work was done by Daniel Scharf, Beçet Açıkmese, Scott Ploen, and Fred Hadaegh of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-44771.

Fast Solution in Sparse LDA for Binary Classification

Special properties of binary classification and greedy algorithms enable speedup.

NASA's Jet Propulsion Laboratory, Pasadena, California

An algorithm that performs sparse linear discriminant analysis (Sparse-LDA) finds near-optimal solutions in far less time than the prior art when specialized to binary classification (of 2 classes). Sparse-LDA is a type of feature- or variable-selection problem with numerous applications in statistics, machine learning, computer vision, computational finance, operations research, and bio-informatics. Because of its combinatorial nature, feature- or variable-selection problems are "NP-hard" or computationally intractable in cases involving more than 30 variables or features. Therefore, one typically seeks approximate solutions by means of greedy search algorithms.

The prior Sparse-LDA algorithm was a greedy algorithm that considered the best variable or feature to add/delete to/from its subsets in order to maximally discriminate between multiple classes of data. The present algorithm is designed for the special but prevalent case of "2-class" or binary classification (e.g. 1 vs. 0, functioning vs. malfunctioning, or change versus no change). The present algorithm provides near-optimal solutions on large real-world datasets having hundreds or even thousands of variables or features (e.g. selecting the fewest wavelength bands in a hyperspectral sensor to do terrain classification) and does so in typical computation times of minutes as compared to days or weeks

as taken by the prior art.

Sparse LDA requires solving generalized eigenvalue problems for a large number of variable subsets (represented by the submatrices of the input within-class and between-class covariance matrices). In the general (full-rank) case, the amount of computation scales at least cubically with the number of variables and thus the size of the problems that can be solved is limited accordingly. However, in binary classification, the principal eigenvalues can be found using a special analytic formula, without resorting to costly iterative techniques. The present algorithm exploits this analytic form along with the inherent sequential nature of